

DNSSEC

Comprendre ses concepts

Yazid Akanho

Fev 2021

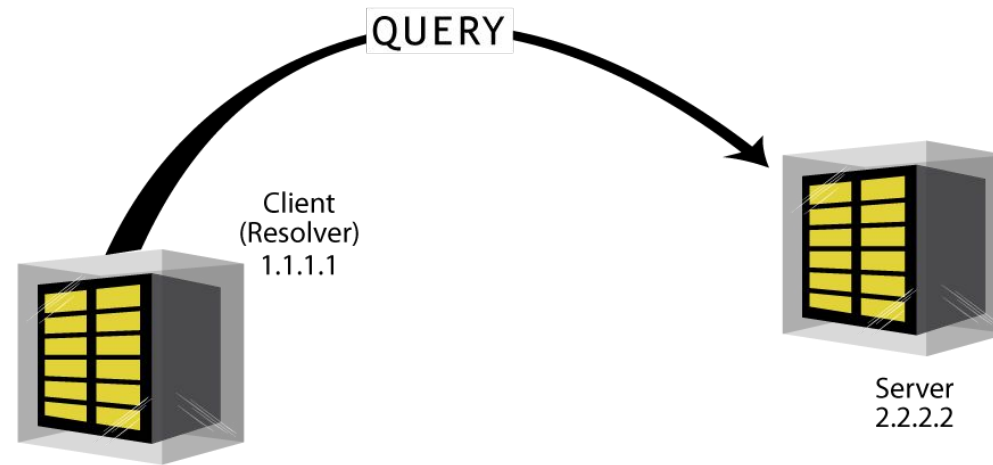


- Un monde sans DNSSEC
Introduction à DNSSEC
- Quelques notions de base de cryptographie
Alors, DNSSEC ... attachez vos ceintures, nous démarrons!
- Plus de détails sur les enregistrements DNSSEC
Validation DNSSEC
- Quelques considérations pour les résolveurs

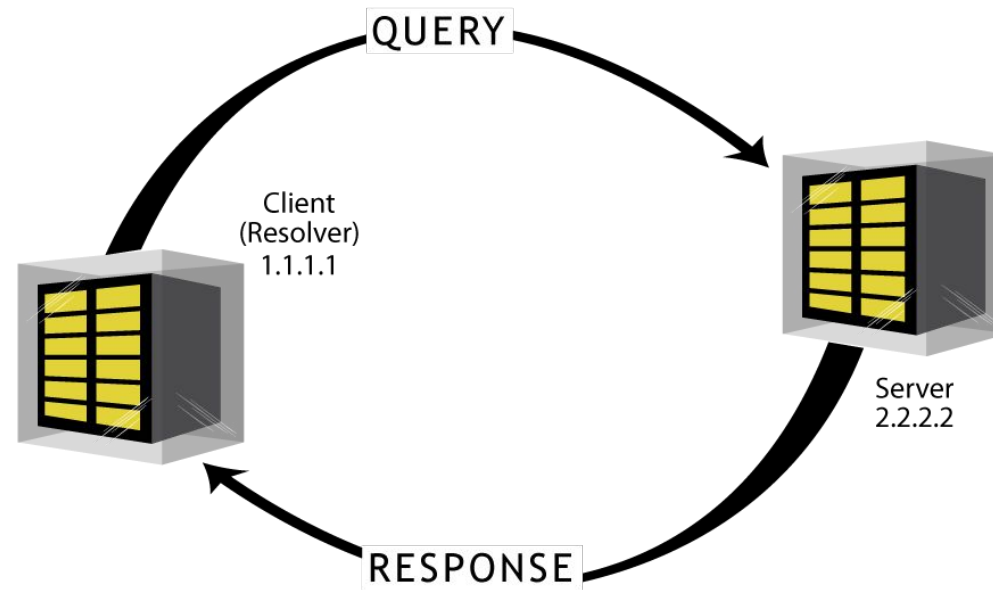
Un monde sans DNSSEC ...



- Un paquet pour une requête DNS, un paquet pour une réponse DNS



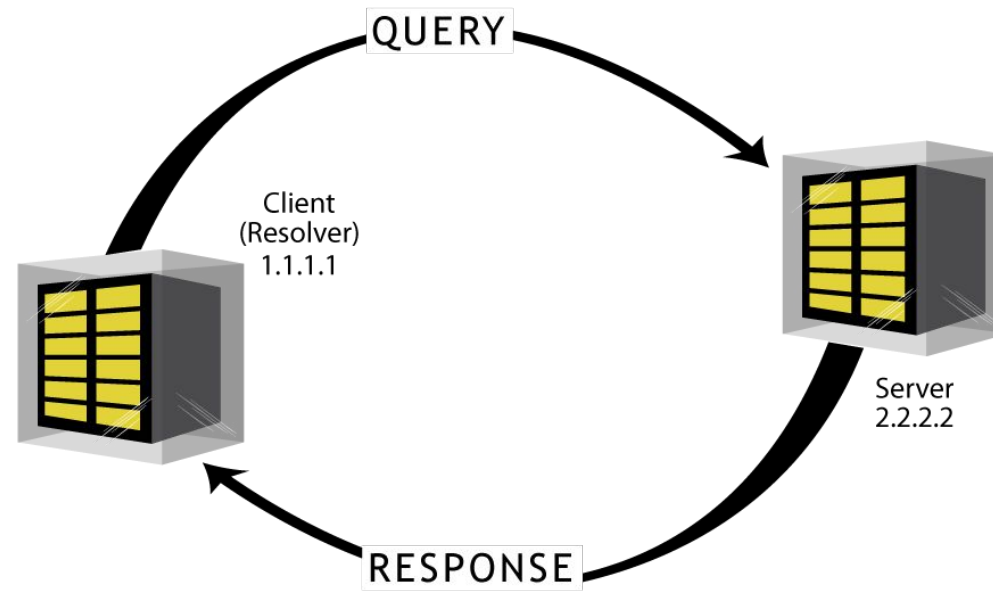
- Un paquet pour une requête DNS, un paquet pour une réponse DNS



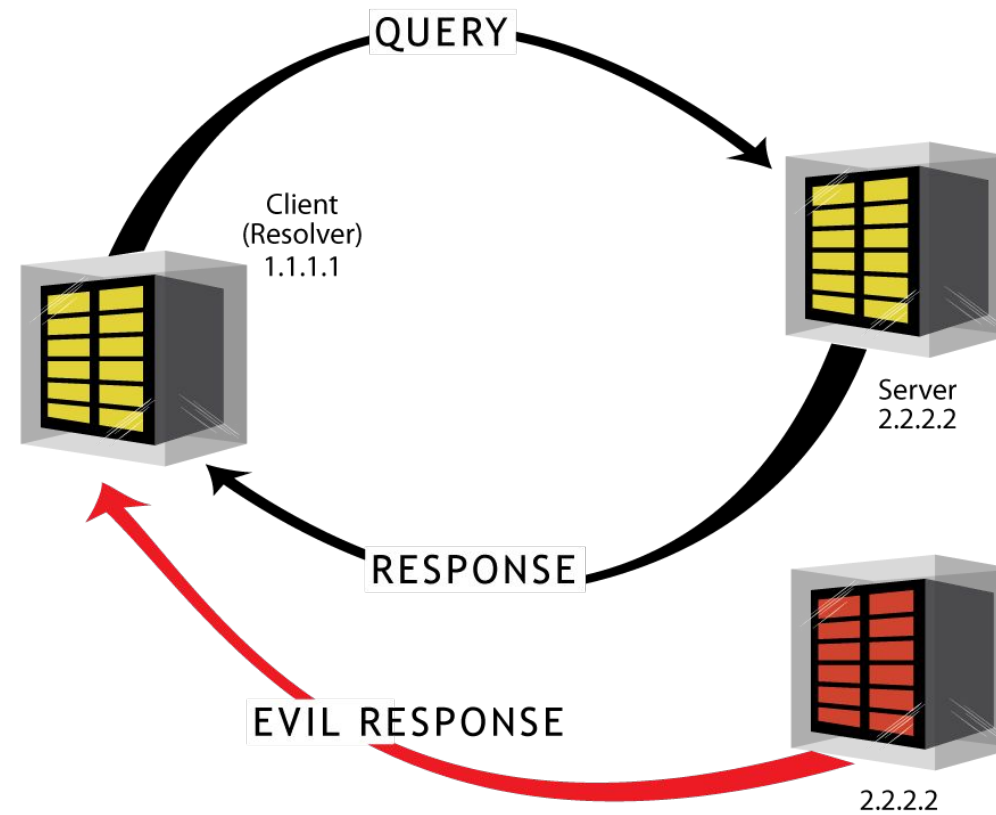
Qui êtes-vous vraiment ?

- Le client doit faire confiance à l'adresse source du serveur
Mais il peut arriver que les adresses source soient truquées ou « usurpées »

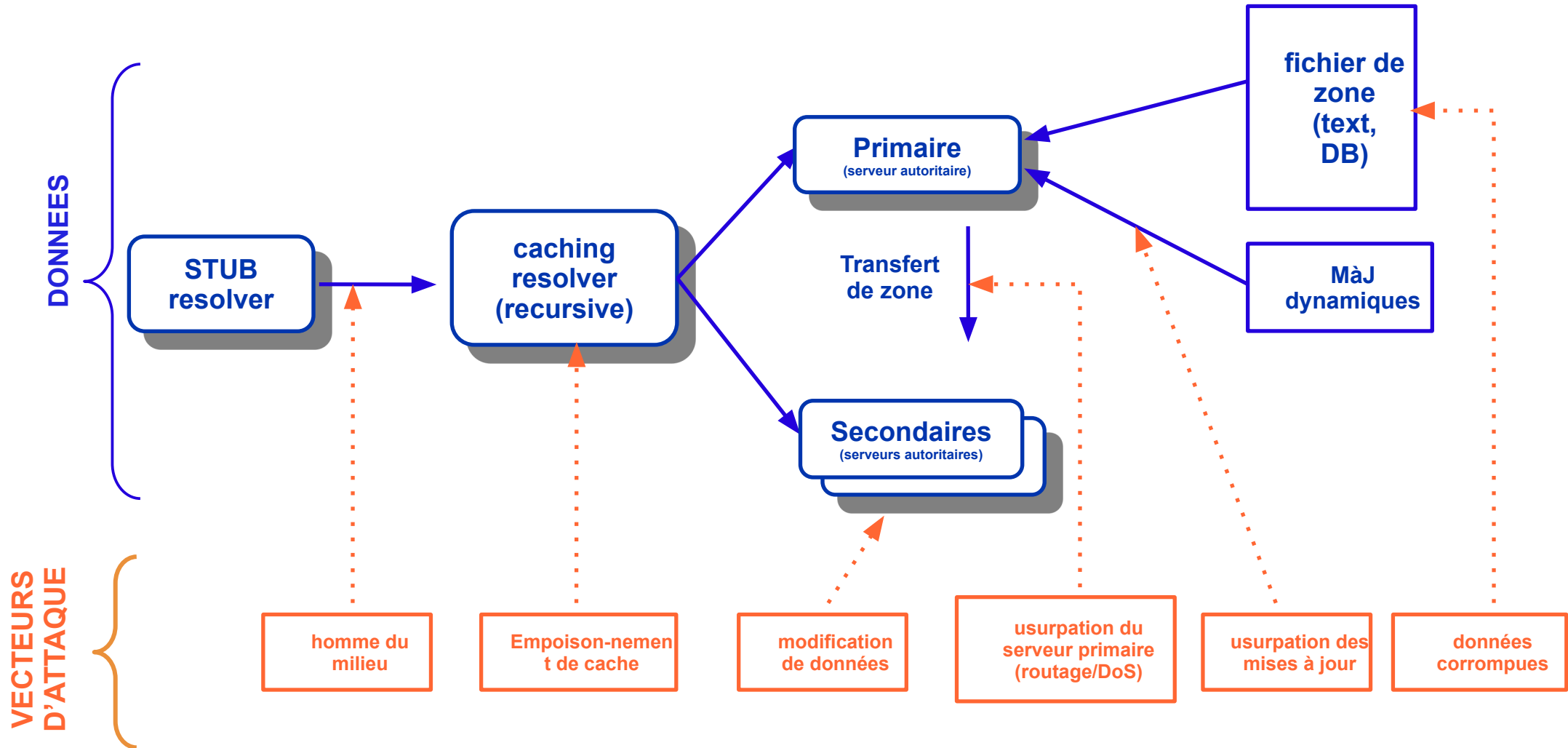
Qui êtes-vous vraiment ?



Qui êtes-vous vraiment ?



Quelles sont les menaces ?



Introduction à DNSSEC



Qu'est-ce que DNSSEC ?

- Extensions de sécurité pour les noms de domaine.
Aide à prévenir l'abus DNS grâce à la cryptographie : fournit l'assurance aux utilisateurs que les données DNS reçues sont **valides** et **authentiques**.
Permettre aux détenteurs de domaines de **SIGNER** leurs données DNS : signez leur zone.
Permet aux opérateurs DNS de **VALIDER** les données DNS passant par les resolveurs DNS.



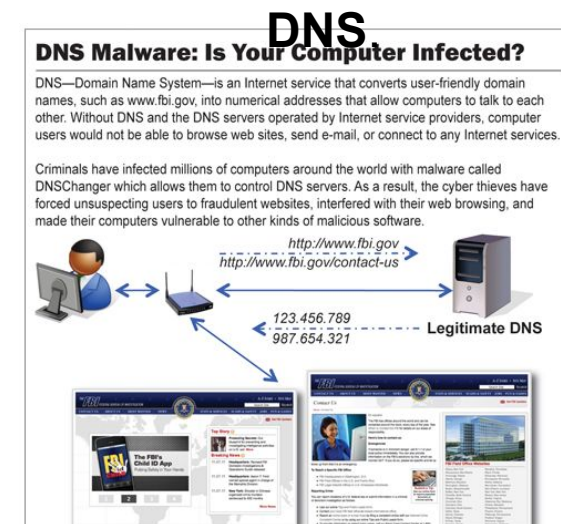
Authenticité : *Sommes-nous certains que l'entité qui publie les données fait autorité ?*

Intégrité : *Les données reçues sont-elles les mêmes que celles qui ont été publiées?*

DNSSEC n'assure ni l'**autorisation** ni la **confidentialité** (chiffrement).

Pourquoi DNSSEC ?

- Les progrès technologiques (processeurs, mémoires, bandes passantes, ...) rendent le DNS de plus en plus vulnérable aux attaques de l'homme du milieu (Man In The Middle).
- Quelques cas célèbres d'attaques DNS dans le monde: Kaminsky (2008), DNSChanger (2011), DNSpionnage (2018), Sea Turtle (2019), ...
- La réputation et la protection des services critiques est devenu un **enjeu majeur de sécurité**: services gouvernementaux, banques, paiements en ligne, ...
DNS Security Extensions (DNSSEC) introduit des **signatures numériques** dans le DNS pour **protéger** cryptographiquement **les réponses**
Avec DNSSEC **entièrement déployé** une entreprise peut être sûre qu'un client obtient des données **non modifiées** (et vice versa).
- Etablit **la confiance** □ on parle de chaîne de confiance (**chain of trust**).



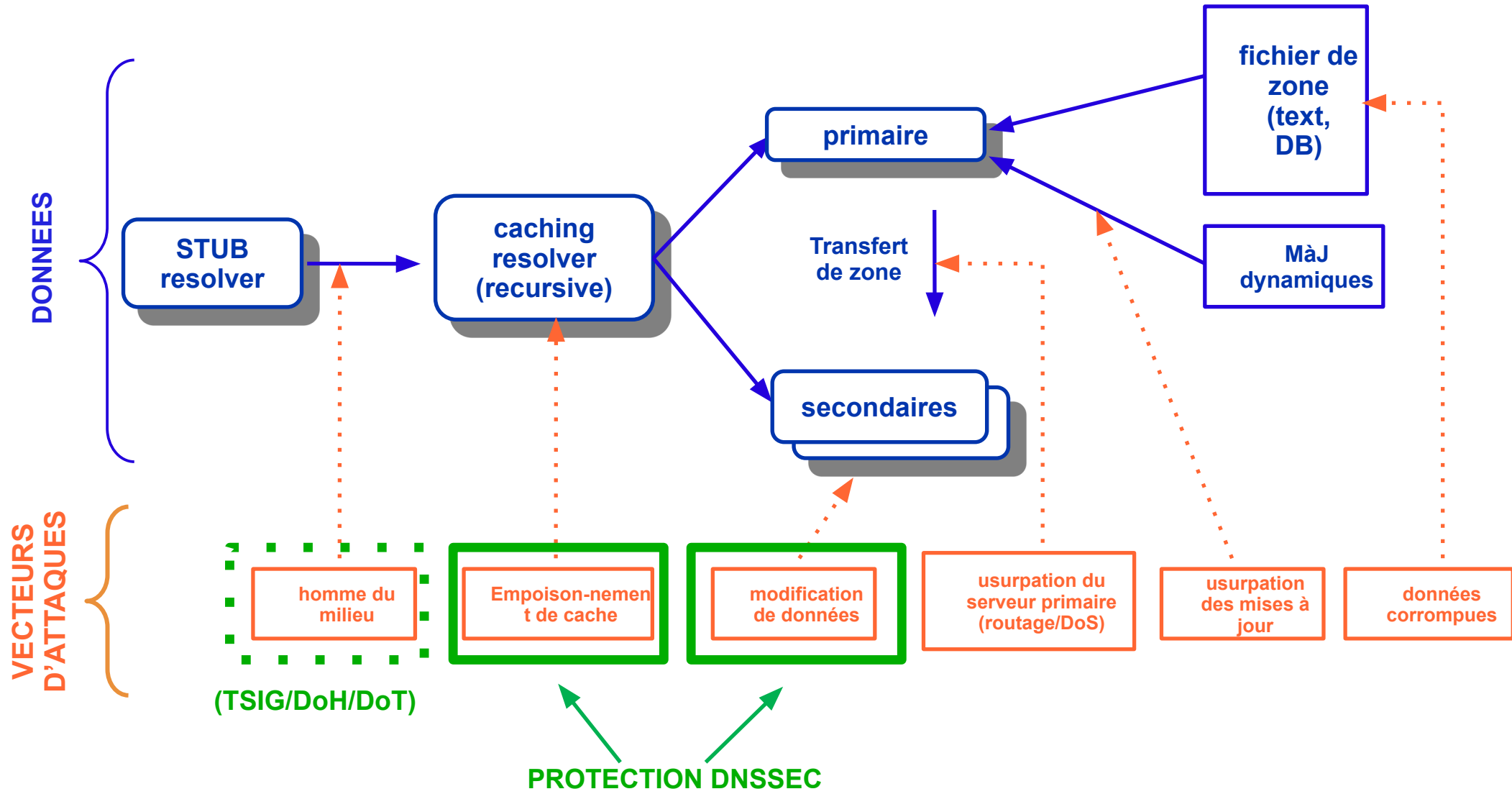
Que fait DNSSEC ?

- DNSSEC utilise la cryptographie à clé publique et les signatures numériques pour fournir :
 - **Authentification** de l'origine des données:
 - « Cette réponse provient-elle vraiment d'un serveur faisant autorité sur la zone example.com ? »
 - **Intégrité** des donnée
 - « Un attaquant (p. ex., l'homme du milieu) a-t-il modifié les données de cette réponse après que les données aient été signées à l'origine ? »
- DNSSEC offre une protection contre l'usurpation de données DNS, et donc contre des attaques comme l'empoisonnement par cache, etc.

Qu'est-ce que DNSSEC ne fait pas ?

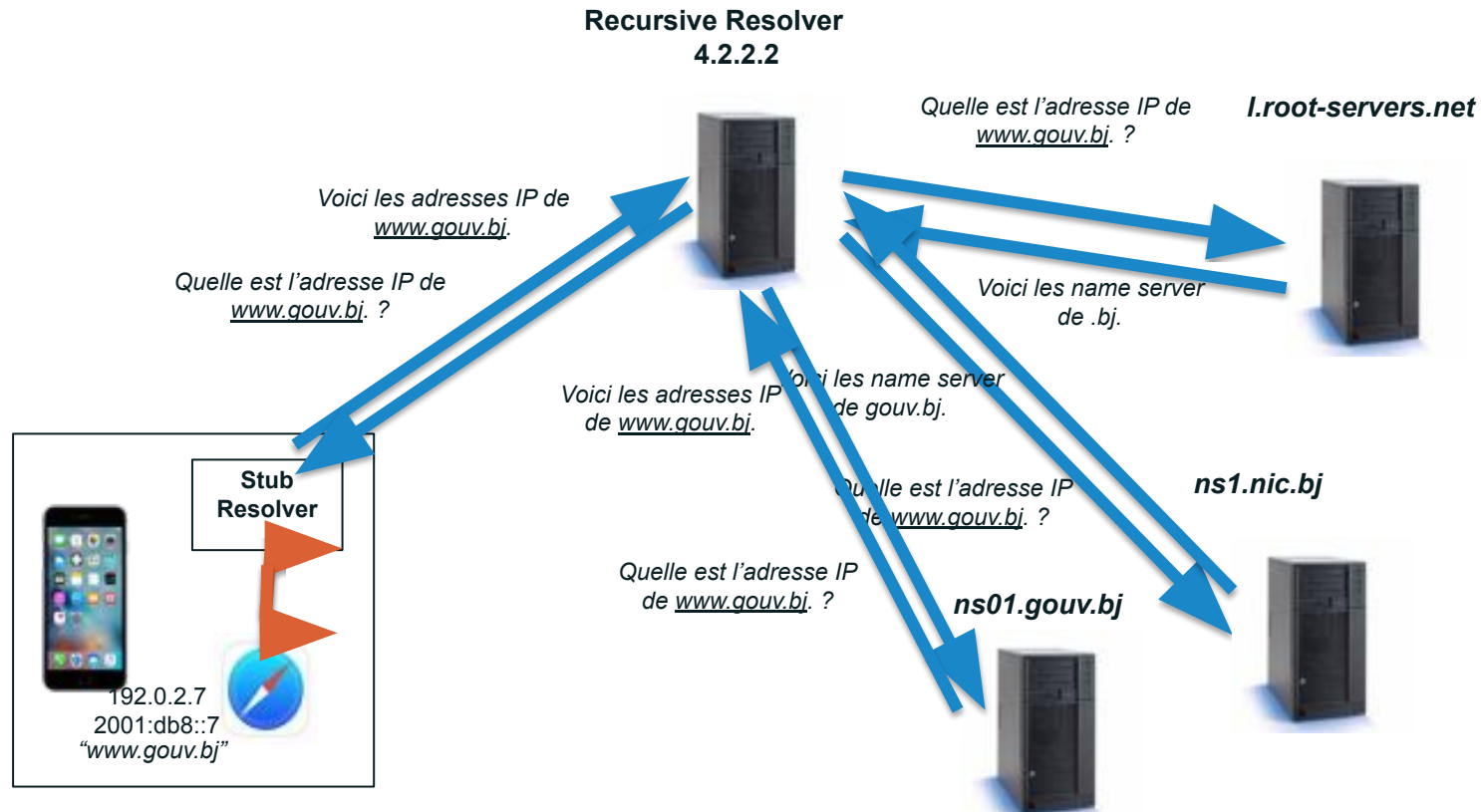
- **DNSSEC ne :**
 - Fournit **aucune confidentialité** pour les données DNS:
 - Pas de chiffrement.
 - Les données transférées seront lisibles par quiconque.
 - **Protège pas des attaques** contre le serveur DNS:
 - DDoS
 - “paquets de la mort”
 - Etc.

Que protège exactement DNSSEC ?



Rappel: processus de résolution de nom DNS

- Un utilisateur tape www.gouv.bj dans son navigateur.
Puis le navigateur se réfère au resolver récursif pour résoudre le nom.
- réponse obtenue aussitôt partagée au navigateur.

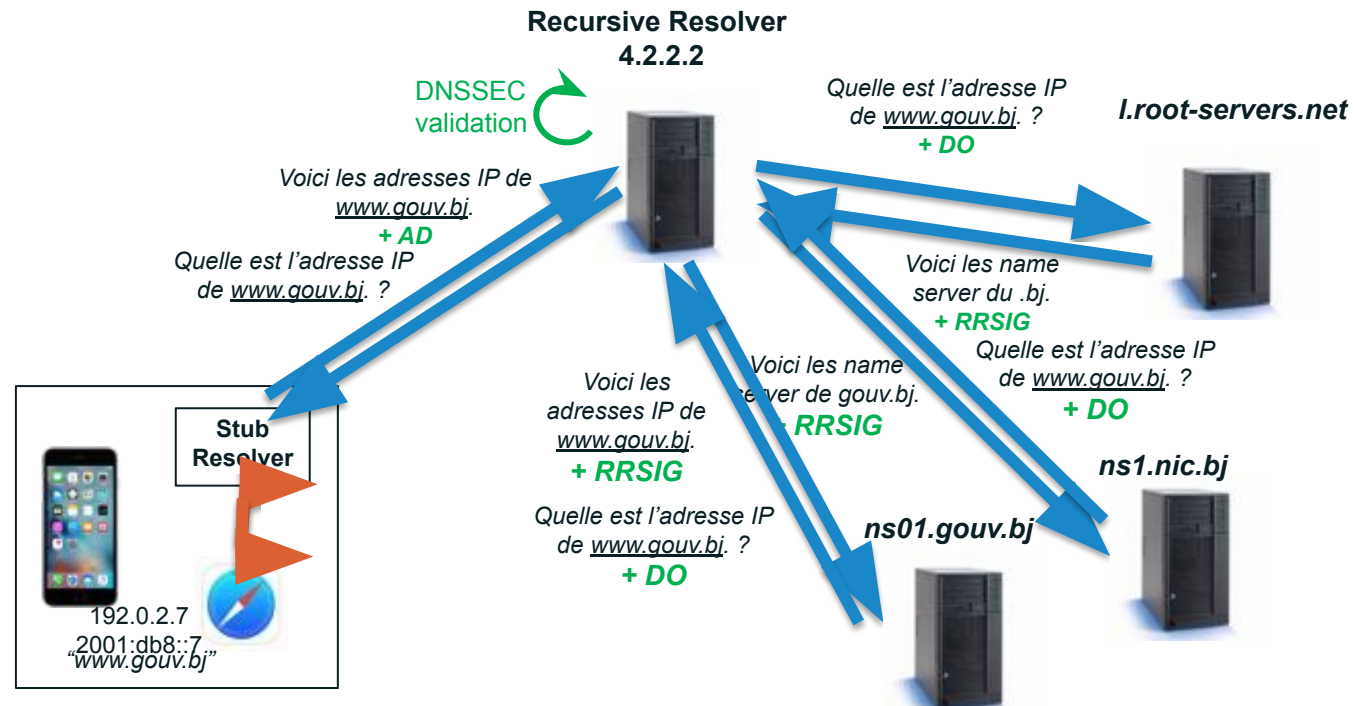


Discussion: comment se fait la résolution DNS avec DNSSEC ?



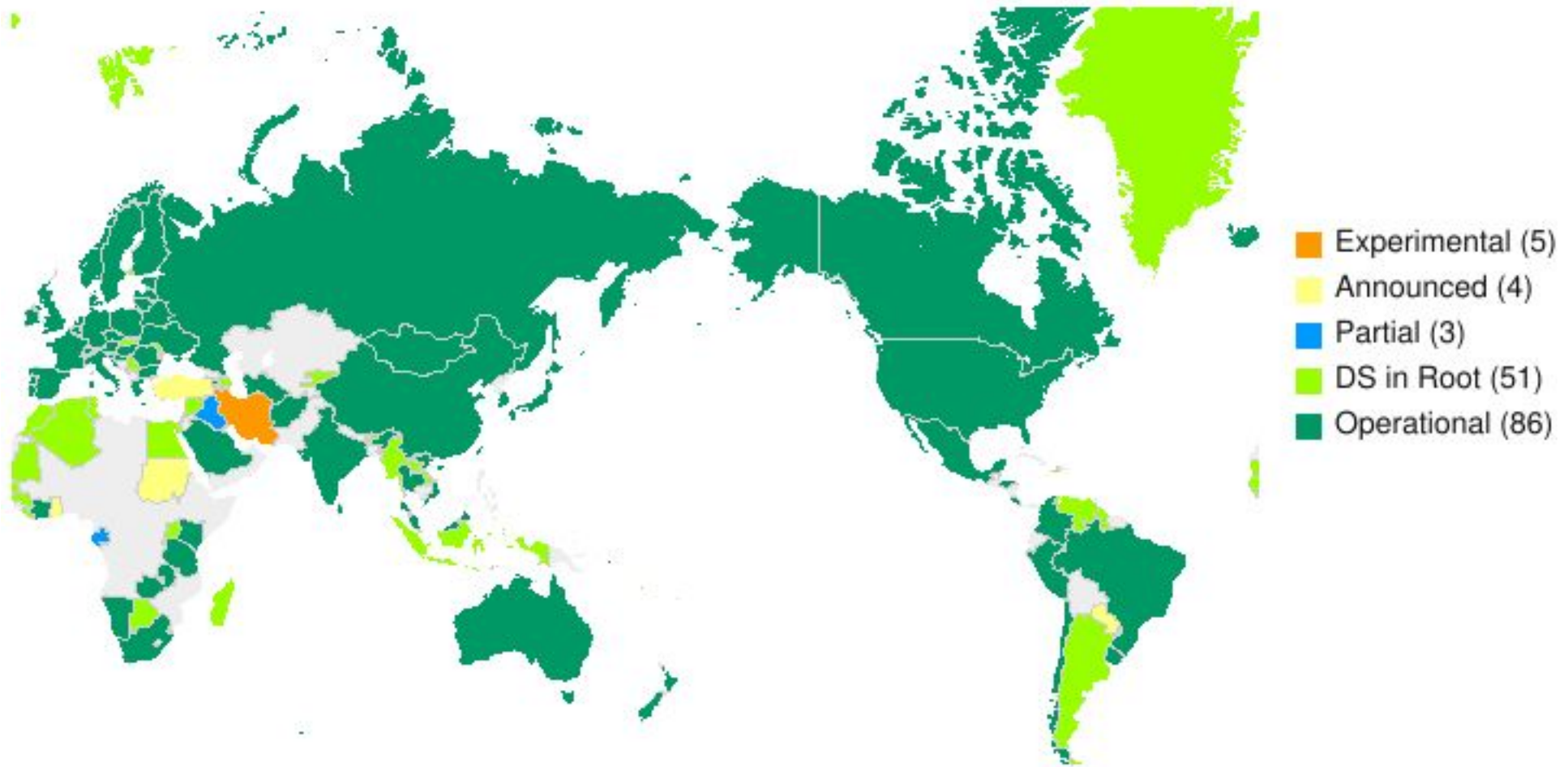
Processus de résolution de nom avec DNSSEC

- Un utilisateur tape www.gouv.bj dans son navigateur.
- Le navigateur se réfère au resolver récursif pour résoudre le nom.
- Le resolver récursif effectue la **validation des données** avant de répondre au navigateur **lorsque la chaine de confiance est établie.**



Statut de déploiement DNSSEC dans les ccTLD du monde

ccTLD DNSSEC Status on 2020-09-14



Qui devrait implémenter DNSSEC ?

- Entreprises: (faire) signer leurs domaines
- Entreprises: activer la validation DNSSEC sur les résolveurs récursifs dans leurs systèmes d'information.
- Opérateurs de Registres (TLD): signer leur zone (.bj, .ci, .ma, .tg, .cd, ...)
- Registrants (détenteur de domaine): (faire) signer leur domaine
- Fournisseurs d'Accès Internet : activer la validation DNSSEC sur les résolveurs DNS de leurs clients/abonnés.
- Hébergeurs: signer et sécuriser les services offerts
- Registraires (vendeurs de noms de domaines): accepter les enregistrements DNSSEC notamment les DS.

Que pouvez-vous faire ?

- **Entreprises**

- Signez vos domaines
Activez la validation DNSSEC sur les resolvers DNS récursifs ou utilisez des resolvers valideurs.

- **Utilisateurs**

- Demandez aux FAI d'activer la validation DNSSEC sur leurs resolvers DNS récursifs ou d'utiliser des resolvers valideurs.

- **Tous**

- Profitez des formations DNSSEC délivrées par des organisations telles que ICANN, ISOC et autres.

Quelques notions de base de cryptographie ...



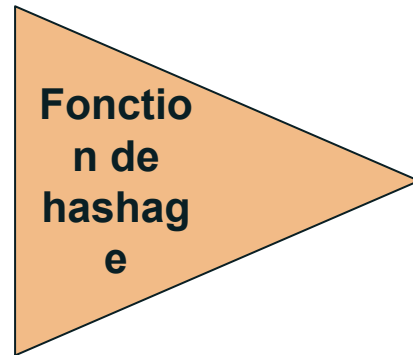
Quelques bases de cryptographie

- Les algorithmes de chiffrement à clé publique fonctionnent avec une paire de clés: une **clé publique** et une **clé privée**.
 - Les données chiffrées avec la clé publique peuvent être déchiffrées avec la clé privée
Les données signées avec la clé privée peuvent être vérifiées avec la clé publique
 - Exemple d'algorithmes à clé publique:
 - Le plus ancien et également plus répandu est RSA
 - Les nouveaux algorithmes basés la cryptographie à courbe élliptique (ECC) tels que: ECDSA, EdDSA et bien d'autres.
- Un algorithme de hachage cryptographique produit une sortie de taille fixe (quelle que soit la taille du fichier d'entrée) appelée **hash** ou **digest**
Deux entrées différentes ne peuvent guère produire le même hash.
 - Le hashage est donc similaire à une « empreinte digitale » du document
Exemple d'algorithmes de hashage : SHA-256, SHA-1 (plus ancien), MD5 (encore plus ancien et obsolète)

Fonction de hachage

- Un algorithme de hachage cryptographique produit une sortie (texte) de taille fixe (empreinte digitale) appelée hash/digest quelle que soit la taille du texte d'entrée.

Texte de taille variable



Texte de sortie de taille fixe

Exemple de digests MD5 (hachage MD5 de 128 bits créé à partir de chaîne de caractères de longueur quelconque) :

One ring to rule them all

Hash

bc713027e780c5d0a8d452b3df9f58dc

One ping to rule them all

Hash

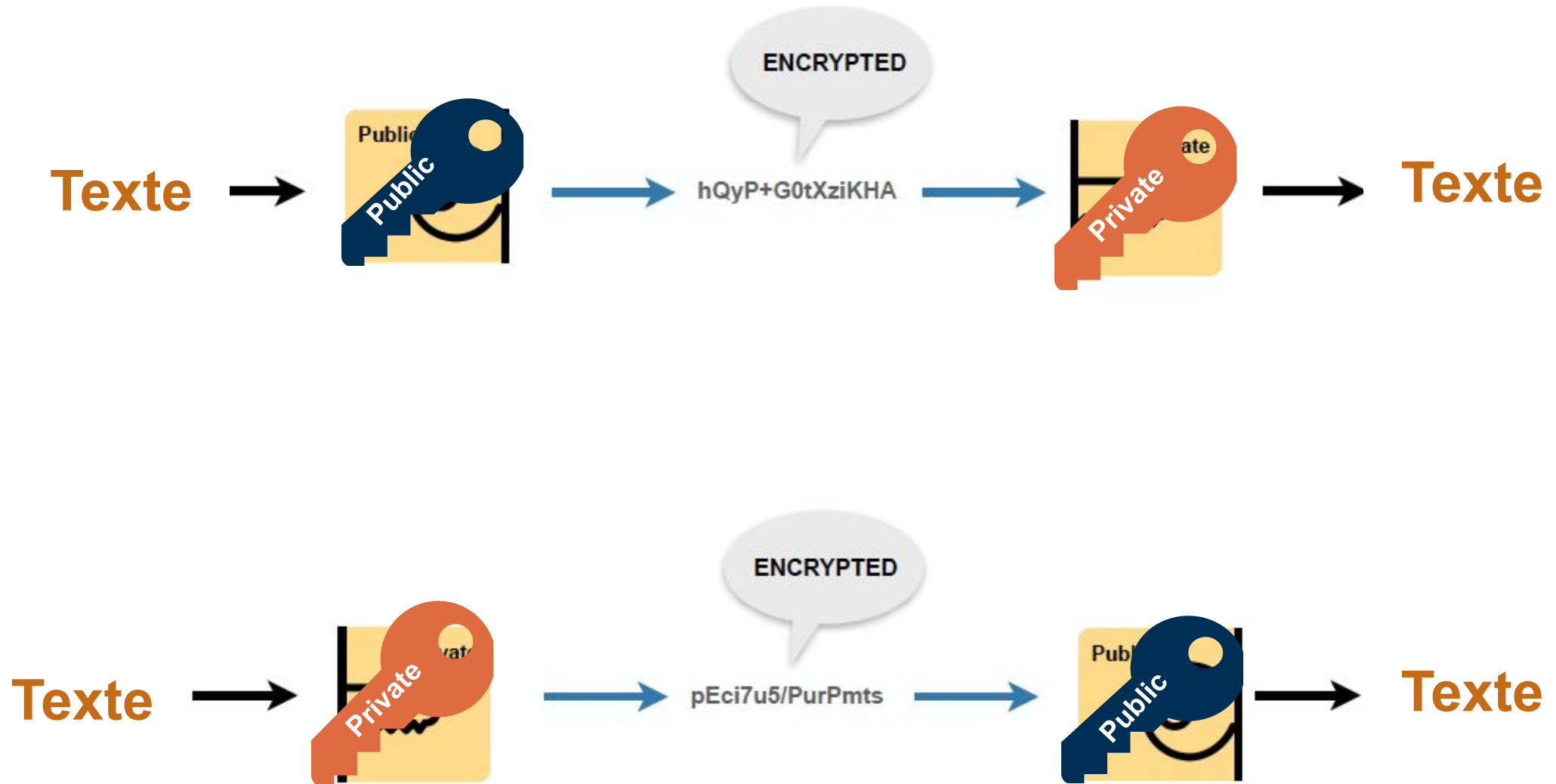
b18d5f6790d95dc29235f3bd2bbf00d7

One ring

Hash

71532c21ac6551759758aaddba2c557a

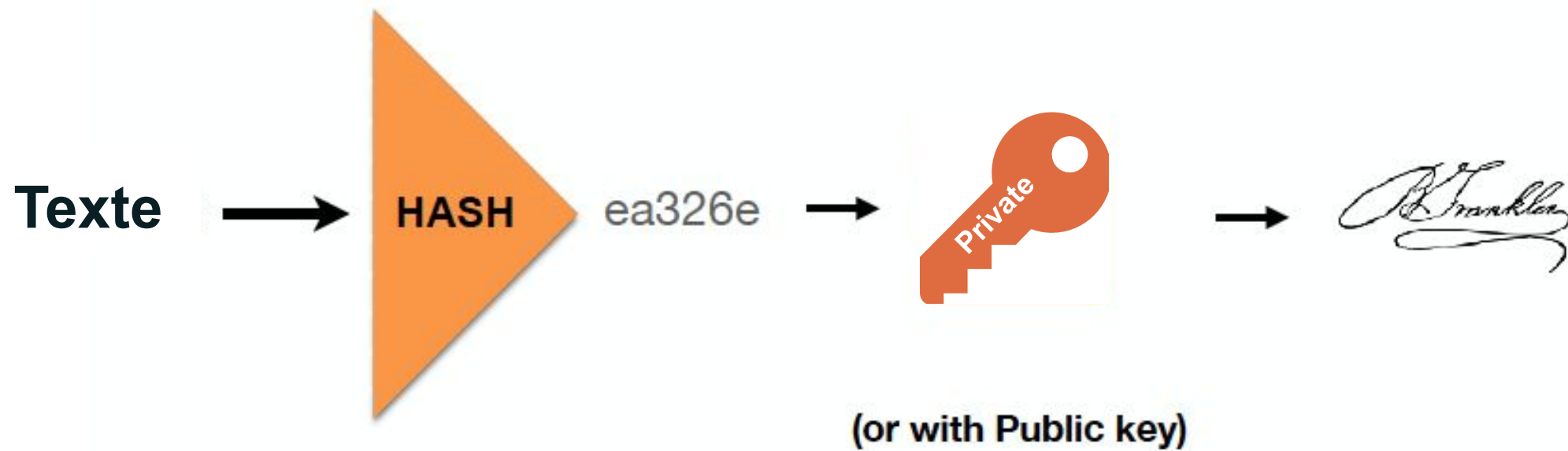
Clés privées et publiques



Signature numérique

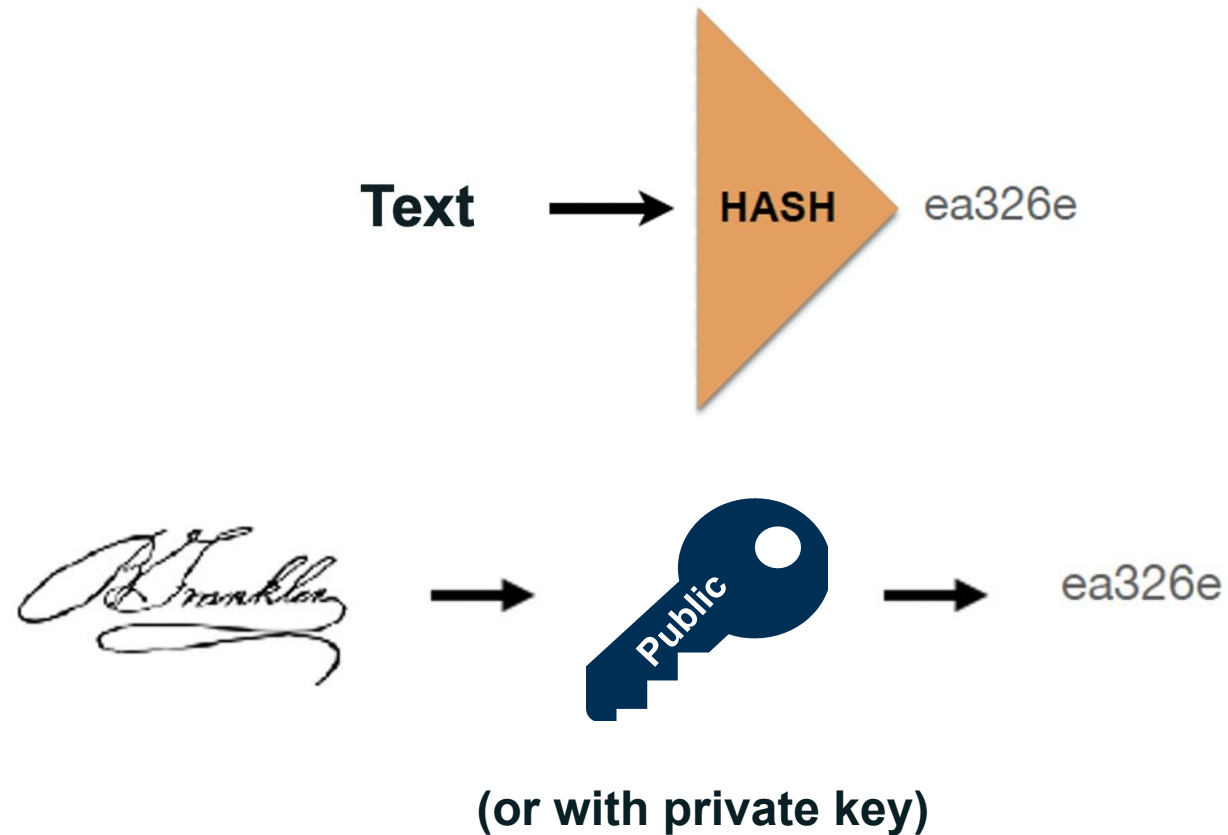
- Nous pouvons combiner le hachage avec la clé privée et publique, pour obtenir la **signature numérique** de n'importe quel texte.

Hash + Chiffrement = Signature Numérique



Signature numérique

Pour vérifier la signature numérique, j'ai besoin du texte et de la clé publique (ou clé privée si elle est signée avec la clé publique)



Alors, DNSSEC !

Attachez vos ceintures de sécurité ...



Calendrier de sécurisation du DNS avec DNSSEC

- 1993: Début de la discussion sur la sécurité du DNS
- 1994: Publication de la première ébauche d'une norme possible
- 1997: Publication du RFC 2065
 - DNSSEC est une norme IETF
- 1999: Publication du RFC 2535
 - Révision de la norme DNSSEC
- 2005: Publication de normes totalement nouvelles
 - RFCs 4033, 4034 and 4035
- Juillet 2010: Zone racine signée
- Mars 2011: zone .com signée
- 2012-: exigences contractuelles pour les nouveaux gTLDs de signer leurs zones
- 2018: Remplacement de la clé KSK sur la zone Racine.

Signer les données DNS

- Dans DNSSEC, chaque zone dispose d'une paire de clés public/privé
Les données de la zone sont signées avec la clé privée
 - La signature des données est généralement dissociée de la fourniture des réponses pour la zone
La conception permet de signer les données **à l'avance** plutôt que « à la volée » pour chaque réponse
- Important : Dans DNSSEC, les données DNS sont signées, pas les messages DNS
 - La signature de messages s'appelle la sécurité des transactions
Un protocole distinct appelé TSIG gère cela

- La clé publique de la zone est publiée dans la zone dans un enregistrement spécifique.
La clé privée de la zone est conservée en toute sécurité
 - Le niveau de protection requis dépend de la façon dont le propriétaire de la zone évalue les risques encourus au cas où la clé privée serait divulguée ou compromise.
- Options pour protéger la clé privée d'une zone :
 - Stockée en ligne sous une forme chiffrée, déchiffrée uniquement en cas de besoin pour la signature des données
 - Le minimum.
 - Stockée hors ligne également sous une forme chiffrée
 - Offre plus de protection.
 - Stocké dans un dispositif physique de sécurité (HSM)
 - Offre le plus de protection, mais un peu exagéré (peut également être coûteux) pour de nombreuses applications.

Rappel des enregistrements de ressources (RR)

- Les données associées aux noms de domaine sont contenues dans les enregistrements de ressources.

- **A** Adresse IPv4
- **AAAA** Adresse IPv6
- **NS** Nom d'un serveur de nom faisant autorité
- **SOA** "Start of authority", apparaît à l'apex de la zone
- **CNAME** Nom d'un alias à un autre nom de domaine
- **MX** Nom d'un « serveur d'échange de courrier »
- **PTR** Adresse IP codée comme nom de domaine (pour la résolution inverse)

DNSSEC ajoute quelques autres:

- DNSKEY
- RRSIG
- NSEC/NSEC3
- DS

Discussion : quels sont les types d'enregistrement pour DNSSEC ?



Nouveaux types d'enregistrement

RRSIG

Resource Records Signature

DNSKEY

DNS Public Key

DS

Delegation Signer
(Chain of Trust pointer)

NSEC

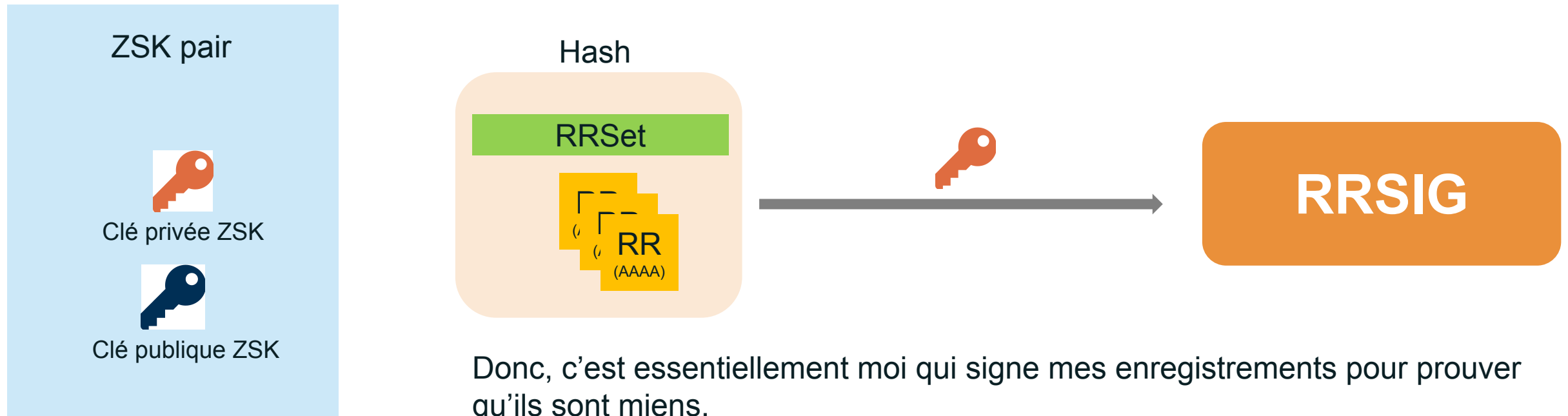
Next Secure

Rappel : Un RRset est un groupe d'enregistrements de ressources de même type et de même nom.

Avec DNSSEC, chaque zone dispose d'une paire de clés de signature de zone appelée ZSK.

ZSK = Zone Signing Key = Clé de signature de la zone.

L'opérateur de la zone crée des signatures numériques pour chaque RRset en utilisant la clé privée ZSK, puis les publie dans la même zone en tant qu'enregistrements RRSIG.



En outre, les opérateurs de zone doivent partager leur clé publique ZSK pour que d'autres vérifient les signatures. Ils publient donc la clé publique ZSK dans un enregistrement DNSKEY dans leur propre zone.

Paire de clés ZSK



Clé privée ZSK



Clé publique ZSK

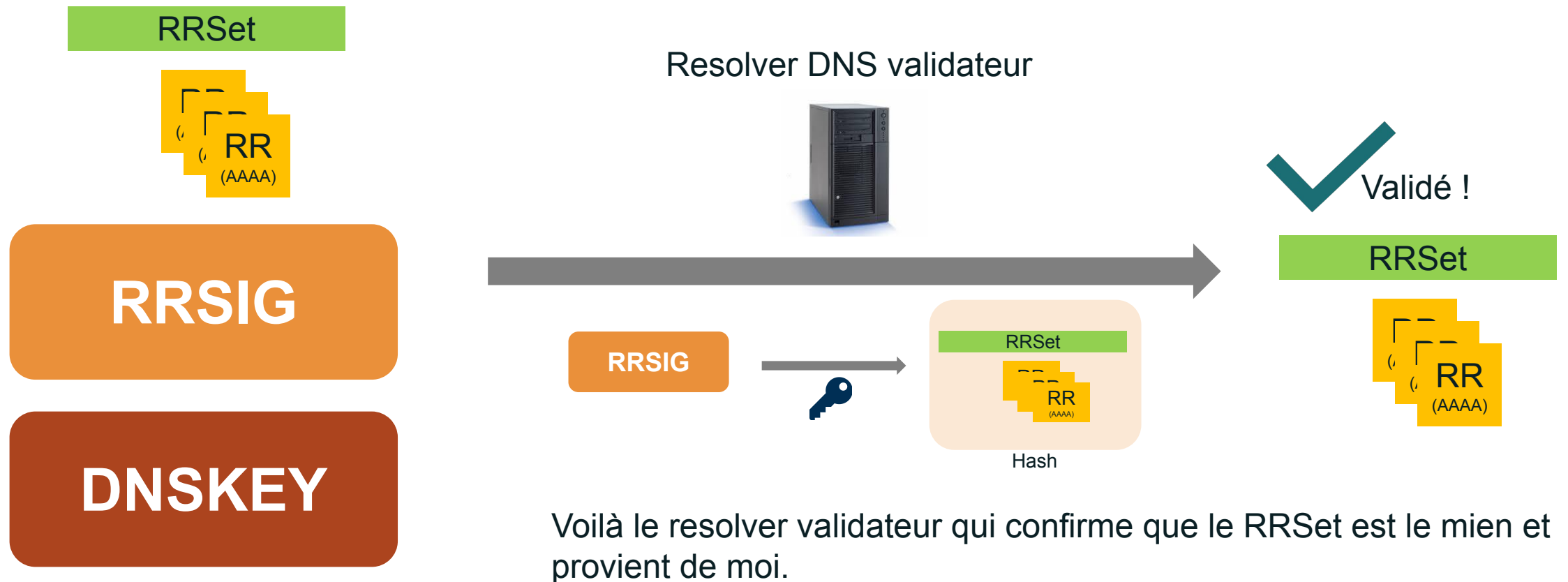


DNSKEY

Donc, c'est essentiellement moi publiant ma clé publique pour permettre aux autres de vérifier mes signatures.

A présent, les résolveurs devraient être en mesure de vérifier les signatures ...

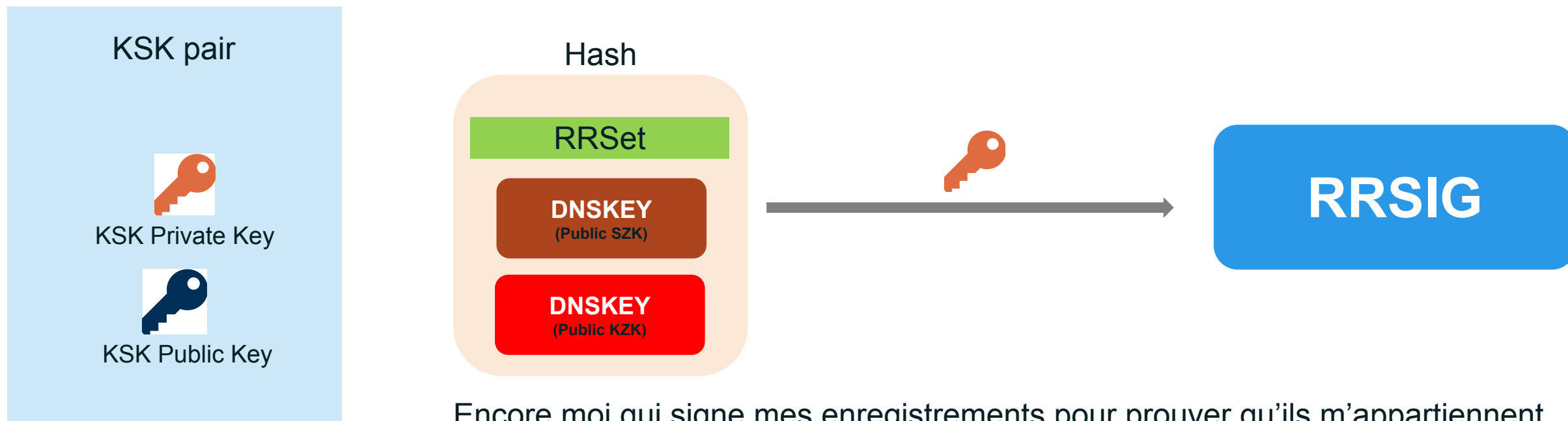
Le resolver récupère l'enregistrement DNSKEY (contenant la clé publique ZSK) du serveur de nom et l'utilise en collaboration avec RRSIG et RRset pour **valider la signature** (RRSIG).



... Donc, tout se résume aux resolvers faisant confiance à la clé publique ZSK obtenue du DNSKEY !

Comment leur faire confiance? (i.e.: comment valider la clé publique ZSK?)

Pour valider la clé publique ZSK, les serveurs de noms disposent d'une autre paire de clé appelée Key Signing Key (KSK) qui fonctionne de la même façon que ZSK: la clé publique ZSK est signée avec la clé privée KSK (la clé privée KSK signe l'enregistrement DNSKEY contenant la clé publique ZSK et la clé publique KSK). La signature obtenue est publiée dans un autre enregistrement RRSIG.



Encore moi qui signe mes enregistrements pour prouver qu'ils m'appartiennent.

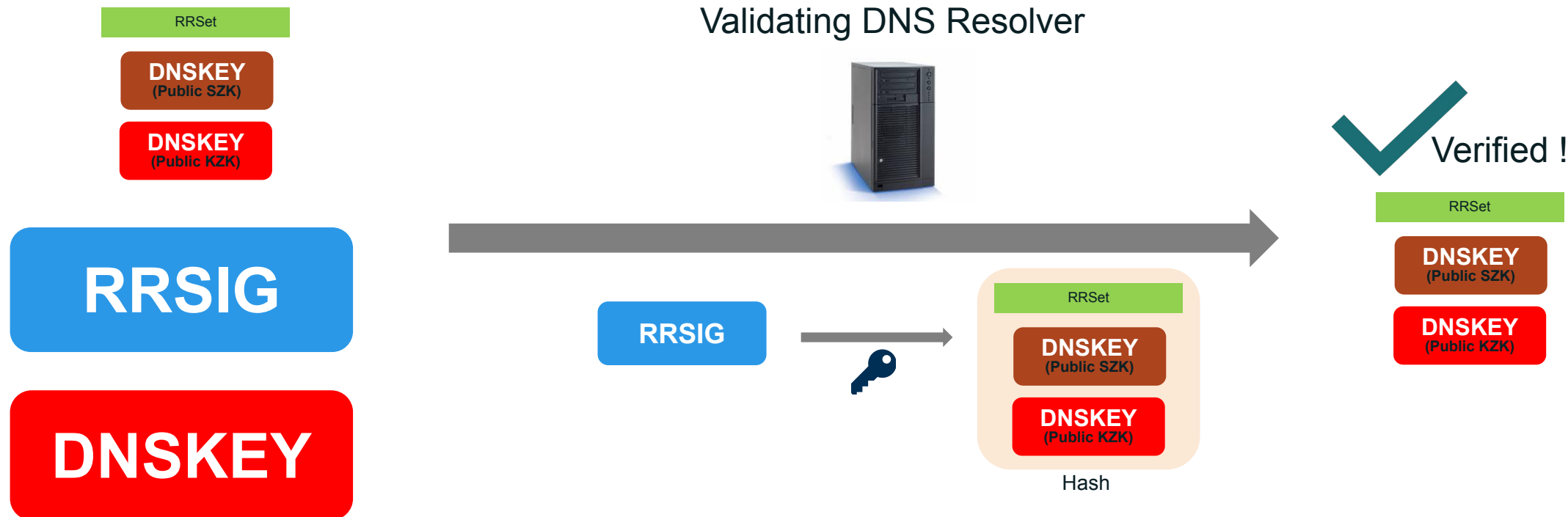
En outre, les opérateurs de zone doivent partager leur clé publique KSK pour que d'autres vérifient la signature. Pour cela, ils publient cette clé publique KSK dans l'enregistrement DNSKEY sur leurs serveurs de noms.



Donc, je publie ma clé publique KSK pour permettre aux autres de vérifier mes signatures.

Les résolveurs devraient à présent être en mesure de vérifier la signature KSK ...

Le resolver récupère l'enregistrement DNSKEY (contenant la clé publique KSK) du serveur de nom et l'utilise en collaboration avec RRSIG et RRset pour valider la signature (RRSIG).



Voilà, le resolver confirmant que la clé publique ZSK est mienne.

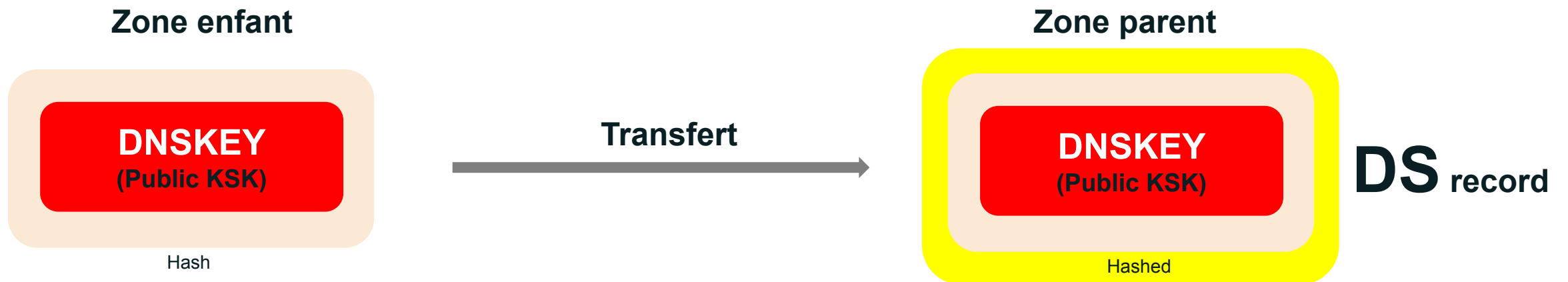
Jusqu'à présent, nous avons établi la confiance dans notre zone. Amusant ???

Mais maintenant nous nous retrouvons avec deux paires clés au lieu d'une ! Pourquoi ?

Changer ZSK est plus facile que de changer KSK; cela permet également d'avoir un ZSK "léger" (par rapport à KSK plus robuste) et donc de réduire la quantité de données échangées entre les serveurs (dans les réponses contenant les clés et les signatures pour chaque RRset).

... En outre, nous devons trouver le moyen de relier une zone avec son parent pour créer la fameuse « Chaîne de confiance » et enfin avoir une clé pour les gouverner tous.

Pour permettre la chaîne de confiance (c'est-à-dire le transfert de la confiance d'un parent à son enfant), le DNS utilise un nouvel enregistrement appelé Delegation Signer (DS).



Chaine de confiance (Chain of Trust)

Lorsqu'un résolveur est redirigé vers une zone enfant (pendant le processus de résolution DNS), le parent fournit également l'enregistrement DS pour cette zone enfant.

De cette façon, le résolveur sait que l'enfant est compatible DNSSEC et a donc un moyen de valider les données de la zone enfant en se basant sur la clé publique KSK: hash de la clé publique KSK de l'enfant est comparé avec le DS fourni par le parent.

Notez que cela nécessite de modifier l'enregistrement DS dans la zone parent chaque fois que le KSK de la zone enfant change. Nous devons donc veiller à faire des changements KSKs d'une manière cohérente pour éviter de briser la zone:

1. Le parent publie le nouvel enregistrement DS à l'avance.
2. Attendre que le DS antérieur arrive à expiration.
3. Supprimer l'ancien enregistrement DS.

Voilà pourquoi changer ZSK est plus facile que de changer KSK.

Chaine de confiance (Chain of Trust)

Finalement, comment pouvons-nous faire confiance à l'enregistrement DS?

Eh bien, nous venons de signer le DS (comme nous l'avons fait avec d'autres Rrsets) en créant un RRSIG correspondant chez le parent.

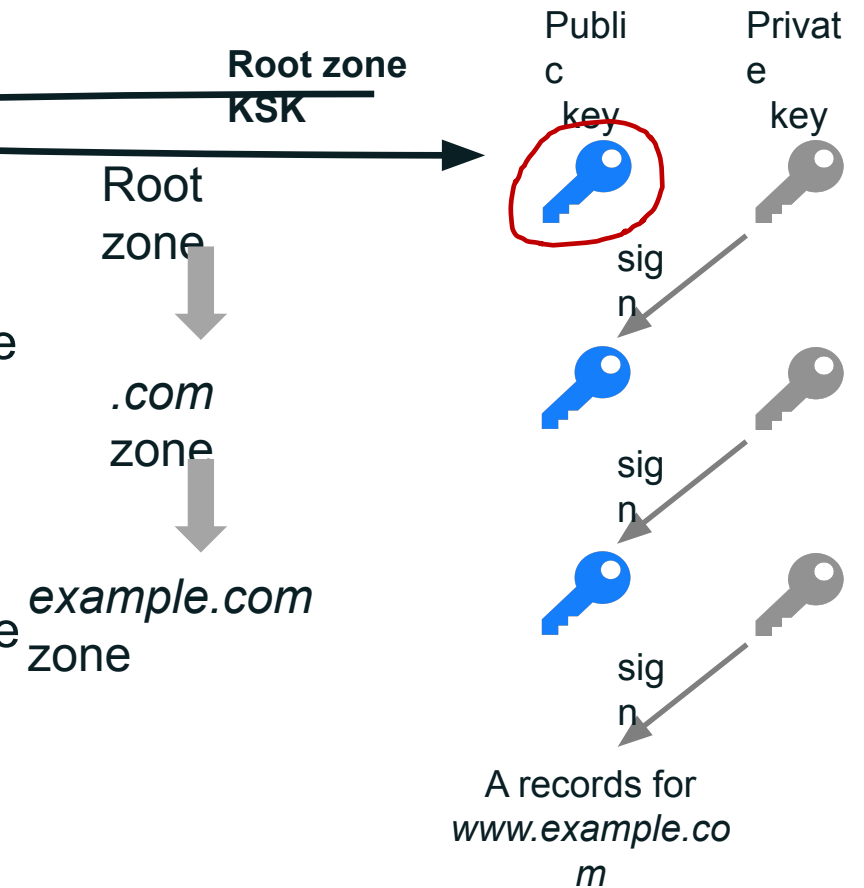
Nous répétons le processus de validation et parvenons à la clé publique KSK du parent ... Puis nous devons remonter à son DS se trouvant chez son parent à lui, et ainsi de suite jusqu'à la racine du DNS.

Finalement, nous allons à la racine et il n'y a pas de parent au dessus! et donc nous devons trouver une solution pour créer un point d'ancrage de confiance pour la clé de la racine.

Ainsi fut mise en oeuvre une solution depuis 2010 appelée:

The Root Signing Ceremony

... à suivre ...



Plus de détails sur les enregistrements DNSSEC



- Champs:
 - **256** ou **257**, *champ à 16-bit*
 - Bit 7 indique une clé DNSSEC
 - Bit 0 indique une clé KSK
 - **3**, l'octet réservé au protocole
 - Sera toujours "3" pour DNSSEC
 - **8**, le numéro correspondant à l'algorithme de signature utilisé (8 = RSA/SHA-256)
 - La clé publique elle-même, encodé en base64
 - Clé RSA à 2048-bit dans cet exemple

```
example.com. 600 DNSKEY 256 3 8 (  
AwEAAAdQdbS3W+EoxaGv21gOGGSUFHB6PNVNC  
PecSLswQ7eKTVtPEYRd+VNDDRZShSOSNFDZq  
eLcO66EO7N8E8udVxGMpBmk59V1YLGAOTIqW  
J5132IGA9JgjSabtYtKU4kMbXqNKM8JrtlJd  
sFF/nixVZzusEl1XZ1u38wozEu0uk39jo5ki  
cju9o5UL2J+cXo7thBY8VRXibmCiz9FWB0G5  
YH/YBgWdI8aFnojoPHbaMUr3G7MObahqCxxv  
41EWPa9AsL97vKi171FD+Jt51Kzq6LcIK55F  
P3I/oUQXGssJ0tINNnR7IVb8uwfo29w5p0DW  
JG930HAItYPDav785Z6Gg8M=  
) ; ZSK; alg = RSASHA256; key id =
```

```
600 DNSKEY 257 3 8 (  
AwEAAAbcTEHTvHv7TzxbeVhFSd9pCivORG73p  
POTsT4WLB7FKtmLwJTXwaKS1bHcY+hm9TL8i  
/H1LcecDEZjm9614I8fk61KwrH9Z7K0ibFrb  
sBirNqXgS43IfRXU1ut4W8BHnOnrKtny2Djd  
KtV46q9nFbzC4WKT/FT0CBGjcc8J5I8SYepO  
J/R2jwFBHdvWnrVKz3tT0ndO0ceuLJOfWyfL  
O/X2GQ6RwWWhojj19V0zpgHockIPtQ+EgSIqx  
unD1Rv07Ezkd/5t1PBJIXjADL9IstSsaol8S  
fgrtyLggM83sWzPTvnnvqGrgQPmqKrnDsLugo  
UPAYIYmgZ7TF2al5BbtZ4T0=  
) ; KSK; alg = RSASHA256; key id =
```

```
www.example.com. 600 A 192.0.2.1
                  600 A 192.0.2.2
                  600 RRSIG A 8 3 600 (
                        20200517225528 20200417225528 47265
```

● Champs:

- **A**, le type d'enregistrements signés
- **8**, le numéro d'algorithme (RSA/SHA-256)
- **3**, le nombre d'étiquettes dans le nom signé
- **600**, le TTL d'origine
- **20200517225528**, expiration de la signature
- **20200417225528**, entrée en vigueur de la signature
- **47265**, l'ID identifiant le DNSKEY devant servir à valider cette signature.
- **example.com**, le nom du signataire (le nom de la zone)
- la signature numérique elle-même, en base64

```
0Od1A6bCmBICLCqQqTpKRFeZrm4Lr/NqXOmg
KuM22cj1l1VLxpdgmwLiU7pTDo2FmOvaNPkgz
a2jhTgSOs6Yj6N0XnkV1e0u2n157YMg26xGv
GqJuPgLKql4KxMjngtdwNB5INQasohALjgAo
uTbu9mQQldyLrkV54P5MUE7lOTFaliWEqW1e
Z/vdaYMc2yKb8CmOQwKxsoWlgnQTYO+lkLuZ
GGffjWH96p6mDbyl5UNA4umSDEqbVKs29Ldv
H7XGOEfkmkze4jSyVUMh57m1DV4ZVLuqx8bQ
YH9zTJPSqvizlSNkuVqssFwknCLwwSOB9FhS
Po9ylhJ9iRPdT34frg== )
```

Prouver que quelque chose n'existe pas!

- Deux types d'«erreurs négatives » dans DNS lorsque l'enregistrement interrogé n'existe pas:
 - Name Error (NXDOMAIN)
 - “No such data” (NOERROR/0)
- Comment prouver cryptographiquement qu'un RRset n'existe pas?
- Pourrait signer des réponses négatives « à la volée”
 - Mais la conception de DNSSEC n'exige pas que la clé privée soit disponible lors du service de la zone
- Ou signez à l'avance : l'enregistrement NSEC

L'enregistrement NSEC

- L'enregistrement NSEC record sert à combler le gap entre deux noms dans une zone
- L'enregistrement NSEC...
 - existe au niveau d'un nom dans un domaine donné
 - Spécifie quels types d'enregistrements existent au niveau de ce nom
 - Renvoie vers le prochain nom dans la zone.
- La notion de “prochain” nom implique un ordre canonique, qui est introduit par le DNSSEC
- Les noms de domaine dans une zone sont ordonnés en:
 - mettant tous les caractères en minuscule
 - Tri des octets non-existants avant « 0 »
 - Tri lexicographique de l'étiquette dans l'ordre décroissant

Une zone avec des enregistrements NSEC

example.com.	SOA	ns.example.com. hostmaster.example.com. 2018041700 3600 600 86400 600
example.com.	NS	ns.example.com.
example.com.	A	10.0.0.1
example.com.	MX	0 mail.example.com.
example.com.	NSEC	east.example.com. A NS SOA MX NSEC
east.example.com.	NS	ns.east.example.com.
east.example.com.	NSEC	ns.east.example.com. NS NSEC
ns.east.example.com.	A	10.0.0.5
ns.east.example.com.	NSEC	ftp.example.com. A NSEC
ftp.example.com.	CNAME	www.example.com.
ftp.example.com.	NSEC	mail.example.com. CNAME NSEC
mail.example.com.	A	10.0.0.2
mail.example.com.	NSEC	ns.example.com. A NSEC
ns.example.com.	A	10.0.0.1
ns.example.com.	NSEC	www.example.com. A NSEC
west.example.com.	NS	ns.west.example.com.
west.example.com.	NSEC	ns.west.example.com. NS NSEC
ns.west.example.com.	A	10.0.0.4
ns.west.example.com.	NSEC	A NSEC
www.example.com.	A	10.0.0.3
www.example.com.	NSEC	example.com. A NSEC

- Le dernier enregistrement NSEC fait la boucle du dernier nom au premier, suivant la liste ordonnée de la zone.
- Chaque enregistrement NSEC possède son enregistrement RRSIG associé.

- Une recherche sur le nom *north.example.com* qui n'existe pas
 - Le serveur répond avec NXDOMAIN et ajoute:

```
mail.example.com.    NSEC    ns.example.com. A NSEC
```

“Aucun nom de domaine entre *mail.example.com* et *ns.example.com* dans la zone”

- Recherche d'enregistrements TXT pour mail.example.com: le nom existe mais n'a pas d'enregistrements TXT
 - La réponse a le code de retour NOERROR, aucun enregistrement dans la section de réponse, et inclut:

```
mail.example.com.    NSEC    ns.example.com. A NSEC
```

“Pas d'enregistrements TXT pour *mail.example.com*, seulement A et NSEC.”

- **NSEC3** est une alternative à NSEC et fournit:
 - Non-énumérabilité
 - Opt-out
- Pourquoi le nom NSEC3?
 - Le nom reflète le nombre de personnes qui le comprennent réellement 😊🤖
C'était une blague.
 - Mais NSEC3 est en effet très compliqué

- NSEC3 ne permet pas d'énumérer la zone ,contrairement à NSEC (“traversée” de la zone),
- NSEC3 procède plutôt par hachage des noms

Exemple:

- Zone: *alpha.example, bravo.example, charlie.example*
- Chaîne NSEC:
 - *alpha.example → bravo.example → charlie.example*
- Chaîne NSEC3:
 - *HASH(bravo).example → HASH(alpha).example → HASH(charlie).example*
 - *ACJENFKS.example → DGJRPFKDM.example → QVNRJVMD.example*

(Note: les “hash” fournis en exemple sont juste à titre illustratif et n’ont rien à voir avec des hash réels NSEC3)

- Standard DNSSEC:
 - Chaque nom dans une zone possède un enregistrement NSEC
 - Y compris les délégations (NS RRsets)
- Opt-Out DNSSEC:
 - Seules les délégations sécurisées ont un enregistrement NSEC3
 - Délégations vers des zones signées
c'est-à-dire les délégations qui ont également un Rrset de type DS.
- Très pratique pour des zones larges telle que *.com*
 - Beaucoup de noms, mais peu de délégations sécurisées
Chaîne NSEC3 beaucoup plus courte que s'il y avait une chaîne NSEC
 - Moins de signatures
Zone signée plus petite (taille)

L'enregistrement DS

- Les champs de l'enregistrement DS:
 - **21700**, key tag/key ID number (de la clé KSK de *example.com*)
 - **8**, the algorithm number (RSA/SHA-256)
 - The DS digest type: **1** is SHA-1, **2** is SHA-256
 - And the digest, in hexadecimal

; This is an excerpt of the .com zone file

```
example.com.      NS      ns1.example.com.
                  NS      ns2.example.com.
                  DS      21700 8 1 (
                    43839D3767944EDD08BA5F342A1F0526FDE1
                    F2E0 )
                  DS      21700 8 2 (
                    7C600DA93B9D0A6EAF8C8DFA9C757D1CC59CD
                    6281EFBAD75DA30FC5B1A121EDC4 )
RRSIG             DS      8 2 600 (
                    20180518010942 20180418010942 22089

                    Lpcx20t+2K3svnR4/KAu7pUtBM90upIeUxF6
                    k7USsg/usvLY2MXmUSTZo00jOD+5CNPMYiLq
                    v/KwDjsxCfjZd25nWy0HLaNCF4kq/Hx7IkA3
                    XxF7c/pjYHSIgGKQ5JdD1x+ns9XNeSxIy7Ic
                    94Gp61SRFd87Mp6KNCbED3BGzmxMTHn4Yql2
                    +TEfvmSHa4shxjtbZOtIFSNnzDKPTwcmtjHK
                    m5WccKUXFrdeGug03TsqJBDWnlzga7NdNITA
                    tWgUKxALyycNGjla4shk6t4mTEpzFe631k2Q
                    0vJamA+MfLZSz6ojT3SU7LyJrMO+RgaslqeE
                    i4UWCs6+JOnLAnFKXQ== )
```


Exemple de zone non signé : example.com

example.com.	SOA	<SOA stuff>
example.com.	NS	ns1.example.com.
example.com.	NS	ns2.example.com.
example.com.	A	192.0.2.1
example.com.	MX	10 mail.example.com.
mail.example.com.	A	192.0.2.2
www.example.com.	A	192.0.1.1
www.example.com.	A	192.0.1.2

Exemple de zone signée : example.com

example.com.	SOA	<SOA stuff>
example.com.	RRSIG	SOA <RRSIG stuff>
example.com.	NS	ns1.example.com.
example.com.	NS	ns2.example.com.
example.com.	RRSIG	NS <RRSIG stuff>
example.com.	A	192.0.2.1
example.com.	RRSIG	A <RRSIG stuff>
example.com.	MX	10 mail.example.com.
example.com.	RRSIG	MX <RRSIG stuff>
example.com.	DNSKEY	<Key that signs the example.com DNSKEY RRset> ; KSK
example.com.	DNSKEY	<Key that signs the rest of the example.com zone> ; ZSK
example.com.	RRSIG	DNSKEY <RRSIG stuff>
example.com.	NSEC	mail.example.com. SOA NS A MX DNSKEY RRSIG NSEC
example.com.	RRSIG	NSEC <RRSIG stuff>
mail.example.com.	A	192.0.2.2
mail.example.com.	RRSIG	A <RRSIG stuff>
mail.example.com.	NSEC	www.example.com. A RRSIG NSEC
mail.example.com.	RRSIG	NSEC <RRSIG stuff>
www.example.com.	A	192.0.1.1
www.example.com.	A	192.0.1.2
www.example.com.	RRSIG	A <RRSIG stuff>
www.example.com.	NSEC	example.com. A RRSIG NSEC
www.example.com.	RRSIG	NSEC <RRSIG stuff>

Discussion : Jouons et récupérons les données du DNSSEC

- Ligne de commande: dig ou nslookup
- Graphique (Web) : <https://www.digwebinterface.com/>

Validation DNSSEC

Resolvers configurés en action ...



- La validation DNSSEC est le processus de **vérification des signatures** DNSSEC
La validation peut se produire au niveau des applications, des resolvers récursifs, ou des stub resolvers (dernières innovations).
- L'essentiel des validations se produit aujourd'hui sur les resolvers récursifs
Que se passe-t-il lorsque la validation échoue ?
 - Mécanisme de signalisation surcharge du resolver recursif au stub resolver.
 - Erreur SERVFAIL, signifie littéralement "je ne veux pas répondre à ta question"
 - Aucun mécanisme de signalisation du stub resolver vers l'application.
 - La plupart des API de resolvers ne sont pas assez fournies pour passer le statut de validation.
 - Résultat des courses: faible expérience utilisateur
- DNSSEC ne garantit pas une bonne réponse; il protège contre l'obtention de mauvaise réponse.

• Ancres de confiance (trust anchors)

- Pour effectuer la validation de signature, vous devez faire confiance à quelqu'un (entité)
Les validateurs DNSSEC ont besoin d'une liste d'ancres de confiance
 - Clés (généralement KSK) qui sont implicitement dignes de confiance
 - Analogue à la liste des CA de confiance dans les navigateurs Web
- Les ancres de confiance ne sont pas détectables
 - Une personne doit prendre la décision de “faire confiance” à une clé.
- L'ancre de confiance la plus importante et la plus utilisée est le KSK de la zone racine du DNS.

Mise à jour des ancres de confiance

- Si une clé change et qu'un validateur a cette clé configurée comme point d'ancrage de confiance, la configuration du validateur doit être mise à jour. La configuration d'ancrage de confiance d'un validateur peut être mise à jour via:
 - Processus manuel
 - Configuration statique
 - Mises à jour automatisées
 - RFC 5011
 - Autre mécanisme de mise à jour fiable
 - Du serveur de nom ou du fournisseur de système d'exploitation

Quelques considérations côté resolver



Quelques considérations matérielles et réseau

- **Mémoire système:** DNSSEC génère des réponses plus longues, l'espace mémoire est donc plus sollicité. Il est recommandé d'effectuer un suivi plus détaillé de l'occupation de la mémoire.
CPU: les validations DNSSEC peuvent augmenter sensiblement la consommation CPU.
- **Interfaces réseau :** pareil que le CPU, en fonction du trafic sur votre réseau.
- **Paquets UDP volumineux:** Certains équipements réseau tels que les pare-feu peuvent avoir des filtres sur la taille des paquets UDP DNS et rejeter les paquets qui semblent trop grand (taille > 51 octets). Vous devez vérifier la configuration EDNS.
Autoriser le port 53 en TCP: cela peut signifier la mise à jour des stratégies de pare-feu ou d'ACL sur les routeurs.

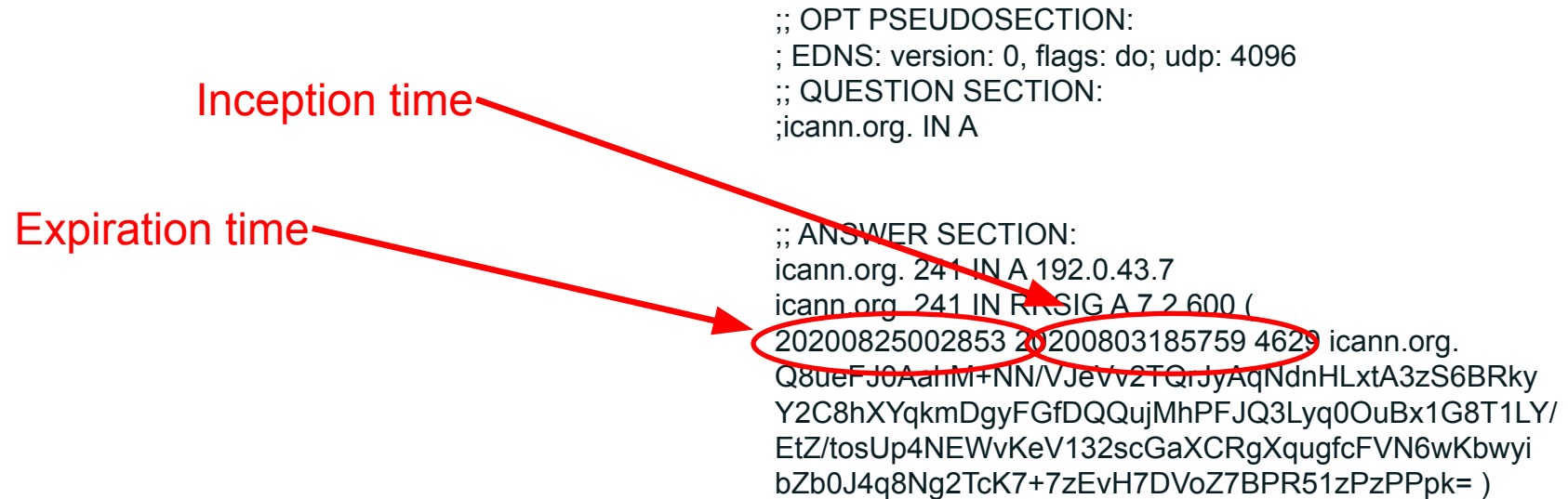
Quelques considérations matérielles et réseau (2)

- **Assurez-vous que le NTP fonctionne correctement:** DNSSEC utilise l'heure et la date du système (serveur) lors de la validation des signatures RRSIG. Donc si la date/heure du résolveur est erronée, ce dernier ne vérifiera pas correctement les signatures; ce qui peut conduire à des attaques. La bonne pratique est d'avoir le resolver utilisant un service NTP de confiance et s'assurer qu'il a toujours la connectivité réseau et est parfaitement synchronisé.

Inception time

Expiration time

```
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags: do; udp: 4096  
;; QUESTION SECTION:  
;icann.org. IN A  
  
;; ANSWER SECTION:  
icann.org. 241 IN A 192.0.43.7  
icann.org. 241 IN RRSIG A 7 2 600 (  
20200825002853 20200803185759 4629 icann.org.  
Q8ueFJ0AahM+NN/VJeVv2TQrJyAqNdnHLxtA3zS6BRky  
Y2C8hXYqkmDgyFGfDQQujMhPFJQ3Lyq0OuBx1G8T1LY/  
EtZ/tosUp4NEWvKeV132scGaXCRgXqugfcFVN6wKbwyi  
bZb0J4q8Ng2TcK7+7zEvH7DVoZ7BPR51zPzPPpk= )
```



Engage with ICANN – Thank You and Questions



One World, One Internet

Visit us at icann.org



[@icann](https://twitter.com/icann)



facebook.com/icannorg



youtube.com/icannnews



flickr.com/icann



linkedin/company/icann



slideshare/icannpresentations



soundcloud/icann